Chapter1

An Introduction to Recommender Systems

1.1 Introduction

- Web은 user가 item에 대한 feedback을 제공할 수 있게 하였다. 이러한 feedback의 form에는 *rating* 혹은 상품의 구매여부 등이 있다. 상품의 구매여부와 같은 form은 user가 노력을 들일 필요가 전혀 없이 data가 수집된다.
- 추천시스템은 이전의 user의 item과 user의 iteraction에 기초한다. 하지만, *knowledge-based recommender systems*의 경우 이전의 데이터 보다는 user-specified *requirements* 에 기초한다.
    - Data가 많을 수록 robust한 예측이 가능하다.
    - Data를 이용한 다양한 추천 모델이 존재한다.
        - Content-based recommender systems : user의 rating과 rating한 item의 특성을 기반으로 한다.
        - Collaborative filtering : 비슷한 user들의 집단을 만들고, 그로부터 missing ratings을 예측한다.(neighborhood models를 포함하는 개념)
        - Knowledge-based systems : user가 그들의 interests를 상호적으로 열거하고, 그 user의 요구사항이 추천을 제공하기 위한 도메인 지식과 결합된다.
        - Advanced models : contextual data(e.g., 시간/장소/social/network 정보 , 외부지식)을 이용하는 모델

1.2 Goals of Recommender Systems

- 추천의 두가지 방법 : 두번째 방법도 첫번째 방법을 이용함으로써 사용할 수 있기 때문에 첫번째 방법이 더 일반적이지만, 많은 경우에 바로 두번째 방법을 푸는 것이 더 쉽다.
    - Prediction version of problem : user와 item의 matrix를 만들고, specified(or observed) 값을 훈련에 사용하고, missing(or unobserved) 값이 훈련 모델에 의해 예측된다. 이러한 문제를 matrix completion problem이라고도 부른다.

    - Ranking version of problems : 특정 user에 대해 top-k items을 결정하거나, 특정 item에 대해 top-k user을 결정한다. 주로 특정 user에 대해 top-k items을 결정하는 방식이 사용되며, 이러한 문제를 top-k recommendation problem 이라고도 부른다.

- 수익증진을 증진하는 더 넓은 *business-centric goal* 성취하기 위한 *operational*하고 *technical*한 목표는 다음과 같다.
    - *Relevance* : user와 가깝게 관련있는 item을 추천하고자 하는 목표. 추천 시스템의 가장 명백한 operational goal.
    - *Novelty* : user가 이전에 보지 않는 item을 추천할 때 더 도움이 될 수 있다. 예를 들어, 인기있는 item은 novel하지 않고, 반복된 인기 item의 추천은 판매의 다양성의 감소를 이끌 수 있다.
    - *Serendipity* : obvious한 추천과 달리 lucky한 발견이다. user를 진심으로 surprising하게 한다는 점에서 단순히 이전에 보지 못한 item을 추천해주는 novelty안 추천과는 다르다. 어떤 user가 특정 유형의 item을 자주 소비하지만, 잠재적인 흥미가 있던 다른 유형의 item을 추천해줌으로써 user를 놀라게 한다. 즉 Indian 요리를 자주 먹는 user에게 가보지 못한 새로운 Indian restaurant을 추천하는 것이 novelty이고, Indian 요리를 아예 알지 못하는 사람에게 Indian 요리를 추천하는 것이 Serendipity이다. Serendipity는 판매의 다양성을 늘리고 user의 흥미의 새로운 trend를 시작할 수 있다. 흥미의 새로운 영역의 발견의 가능성은 long-term 하고 strategic한 이익을 가진다. 관련성이 없는 추천을 하는 short-term한 단점이 있지만, 많은 경우에 이러한 이익을 더 중요시 한다.
    - *Increasing recommendation diversity* : 추천하는 top-k item의 종류를 다양하게 함으로써, 최소 한개 유형의 item에는 흥미가 있을 가능성을 높인다. 만약 모두 비슷하다면 그 중 어떠한 item도 마음에 들어하지 않을 가능성이 있다. 이러한 다양성은 user가 비슷한 item 추천의 반복에 지겨워하지 않는 이점을 가져온다.

- 실제 추천시스템에서는?
    - 추천되는 상품의 유형은 굉장히 다양하다.

Table 1.1: Examples of products recommended by various real-world recommender systems

System	Product Goal
Amazon.com [698]	Books and other products
Netflix [690]	DVDs, Streaming Video
Jester [689]	Jokes
GroupLens [687]	News
MovieLens [688]	Movies
last.fm [692]	Music
Google News [697]	News
Google Search [696]	Advertisements
Facebook [691]	Friends, Advertisements
Pandora [693]	Music
YouTube [694]	Online videos
Tripadvisor [695]	Travel products
IMDb [699]	Movies

- 전통적인 추천 외에도, search results를 기반으로 추천하는 *computational advertising*, SNS에서의 친구 추천은 *structural relationships*을 이용한 *link prediction,* 마지막으로 reciprocal recommender과 같은 것들이 있음.

1.3 Basic Models of Recommender Systems

1.3.1 Collaborative Filtering Models

- sparse한 rating matrices에 의존해야 하는 것이 main challenge이다.
  - "specified"/"observed" rating과 "unobserved"/"missing" 으로 이루어져 있는데 missing rating이 훨씬 많음
- observed rating은 다양한 users와 items간에 많은 관계가 있기 때문에 missing ratings을 추측할 수 있다. 따라서 대부분 cf는 item간의 상관관계 혹은 user간의 상관관계, 혹은 그 둘다에 집중하여 observed된 rating으로 missing rating을 예측하는 과정을 거친다. 이 과정에서 자주 사용되는 두 가지 주요 방법이 있다.

  1. *Memory-based methods* : neighborhood-based collaborative filtering algorithms이라고도 불린다. neighborhoods에 기초한 cf이고, 이러한 neighborhoods는 두가지 방법으로 정의될 수 있다.
     - *User-based cf* : matrix의 *rows*를 사용하여 user간의 similarity를 계산하고, similarity가 높은 user의 rating을 이용하여 추천.
     - *Item-based-df* : matrix의 *columns*를 사용하여 item간의 similarity를 계산하고, similarity가 높은 user의 rating을 이용하여 추천.

     → memory-based techniques는 실행하기 간단하고, 결과를 해석하기 쉽다. 하지만, sparse ratings matrices에서는 잘 작동하지 않아서 특정 user/item의 경

우 예측이 어려울 수 있다. 즉, sparse rating matrices의 rating 예측에서 full *coverage*가 부족하다. 예를들어 해당 user와 비슷하고, 어떤 item에 rating한 user를 찾는 것은 어려울 수 있다. 하지만, top-k items이 필요한 경우에는 문제가 되지 않는다.

2. Model-based methods : machine learning and data mining methods(e.g. decision trees, rule-based models, Bayesian methods and latent factor models)이 사용. latent factor models같이 이러한 모델들의 대부분은 sparse ratings matrices일지라도 높은 coverage를 가진다.

→ 최근에는 두 방법의 조합해서 사용한다.

1.3.1.1 Types of Ratings

Figure 1.1: Example of 5-point interval ratings

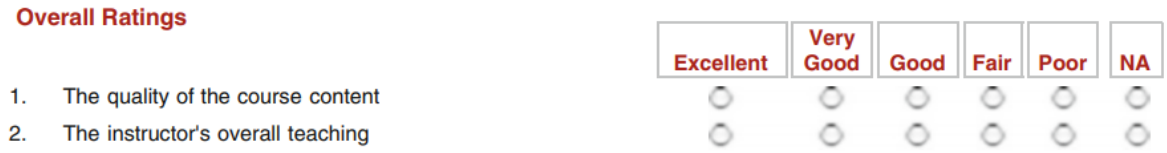


Figure 1.2: Example of ordinal ratings used in Stanford University course evaluations

- rating을 연속적/이산적으로 할 수도, ordinal/categorical/binary values values를 사용할 수도 있다.
- likely와 dislikely의 rating개수가 동등하지 않은 *unbalanced rating scale*를 사용할 수도, 짝수 개의 rating을 사용하여 neutral rating을 없애버린 *forced choice rating system*을 사용할 수도 있다.
- *explicit* 이나 *implicit* feedback, 혹은 둘 다를 추천시스템에 적용할 수도 있다.

    - *explicit feedback* : user가 분명하게 rating한 것. observed rating은 *preferences*에 대응한다. ex) 별점
    - *implicit feedback* : user가 분명하게 rating하지 않은 내재적인 rating. observed rating은 *confidence*에 대응한다. ex) 브라우저에서 user의 행동
- ***unary rating*** : user와의 affinity(관련성)만을 나타내고, 그들의 불호는 보이지 않는 메커니즘. implicit dataset을 활용하는데, 이것은 *positive preference utility matrix*라고 불려진다. 오직 positive preference만 알 수 있기 때문이다. ex) 클릭, 구매, 좋아요
    - user가 특정 행동을 취하지 않을 것으로 가정한다.
    - missing entries를 0으로 초기화하고, 최종 예측값이 0에서 가장 먼, 가장 큰 값인 항목을 기반으로 추천을 한다. 만약 0으로 초기화하지 않으면, observed와 unobserved data간에 큰 구별이 사라져서 overfitting이 발생할 수 있는 문제가 있다. 오히려 0으로 초기화함으로써 unary rating의 가정이 있음에도 불구하고 좀 더 bias가 생겨 overfitting을 감소할 수 있다.
    - explicit feedback에서는 missing value를 어떤 값으로 대체하게 되면, 큰 bias가 발생할 수 있기 때문에 대체하지 않는다.

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ordered ratings

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Unary ratings

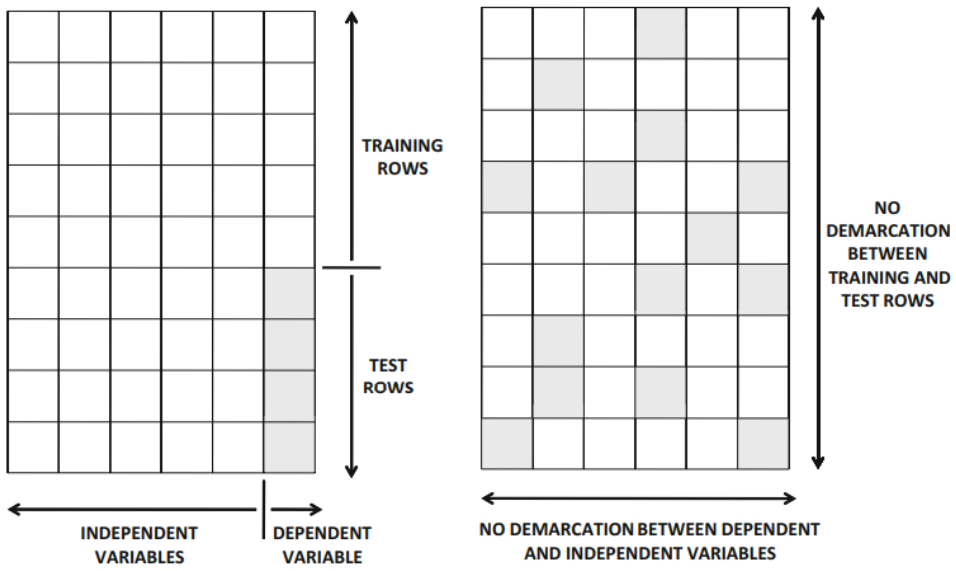
Figure 1.3: Examples of utility matrices

1.3.1.2 Relationship with Missing Value Analysis

- cf는 missing value analysis와 매우 밀접하다. 대부분의 missing value analysis의 방법을 사용할 수 있다. 하지만 그 중엔 매우 large하고 sparse한 matrix에 적용하는 문제에 맞춰서 사용해야 하는 경우도 있다.

1.3.1.3 Collaborative Filtering as a Generalization of Classification and Regression Modeling

- cf metghods는 classification과 regression modeling의 일반화로 볼 수 있다.
- independent/dependent variables 와 train/test set의 구분이 columns/rows가 아닌 entries를 기준으로 존재한다.
    - 어떤 columns에도 missing value가 있을 수 있기 때문에 independent/dependent variables의 구별이 따로 있지 않다.
    - 어떤 rows에도 missing value가 있을 수 있기 때문에 train/test set의 구별이 rows를 기준으로가 아닌 entries 기준으로 존재한다.
- matrix completion 문제는 data가 mxn의 matrix로 통합되어있다. 따라서 test data가 train 과정에서 섞여 들어가기 때문에, 새로운 data에대한 예측력이 낮은 *transductive* setting 이다. 즉, 이미 구축된 cf model로 새로운 user/item에 대한 예측은 정확도가 낮을 수 있다. model-based cf에서 이것은 특히 더 문제이다. 하지만 최근 몇 matrix completion models은 *inductive*하도록 디자인 되었다.
    - transductive : out-of-sample에 대해 잘 예측하지 못하는 것
    - inductive : out-of-sample에 대해 잘 예측하는 것



(a) Classification (b) Collaborative filtering

Figure 1.4: Comparing the traditional classification problem with collaborative filtering. Shaded entries are missing and need to be predicted.

1.3.2 Content-Based Recommender Systems

- item의 content attributes로 추천을 한다. 각 user가 rating한 item의 content attributes가 train data의 독립변수, rating이 종속변수로 사용되고, 이를 통해 *user-specific* model을 만든다.
- cb의 장단점
  - 장점 : 새로운 item이어서 rating이 충분하지 않거나, 심지어 아예 존재하지 않다고 할지라도 그 것의 content attributes를 통해 추천을 할 수 있다.
  - 단점
    1. *obvious* recommendation만을 제공한다. 즉, 소비하지 않은 새로운 item은 추천되지 않고, 비슷한 item만 추천된다.
    2. 새로운 user에 대한 추천의 정확성이 낮다. 새로운 user의 history를 사용하기 때문에, 해당 data가 적다면 추천이 잘 이루어지지 않고, 소수의 data에 overfitting될 것이다.

1.3.3 Knowledge-Based Recommender Systems

- rating을 사용하기가 힘든 상황에서 item을 추천하는데 특히 더 유용한 방법이다. 신규 item이거나 많은 업데이트로 이전과 모델이 많이 달라진 경우, 해당 item에 다양한 options(preperties)이 존재하는 경우엔 rating을 이용하기가 쉽지 않다. (options의 조합과 같은 특정 예시에 대한 충분한 rating을 모으기는 쉽지 않음)

- 고객의 요구사항과 item의 content 사이의 유사성에 근거하여 추천을 한다. 즉, cb/cf의 경우와는 달리 오로지 user가 *explicitly specify what they want*하게 하고, 그것만을 이용한다.

Table 1.2: The conceptual goals of various recommender systems

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge

- Knowledge-Based Recommender Systems은 어떤 knowledge를 사용하느냐에 따라 분류된다.

  → 두 방법 모두 user가 요구사항을 변화시키며 추천 precess를 반복한다는 점은 같다. 하지만, Constraint-based systems은 item을 search하기 위해 rules(or constrains)를 사용하지만, Case-based systems은 target cases와 similarity metrics의 조합을 사용한다는 점이 다르다.

  1. *Constraint-based recommender systems* : user들은 요구사항/제약사항(e.g., lower or upper limits) 을 기술한다. 이를 item attributes와 매치하는 과정에서 domain-specific rules(e.g., “Cars before year 1970 do not have cruise control.”) 과 user의 attributes와 관련된 rule(e.g., “Older investors do not invest in ultra high-risk products.”) 을 사용할 수도 있다. 그 후, user들은 추천 결과를 보고 요구사항을 수정하는 것을 상호적으로 반복하며 탐색한다.

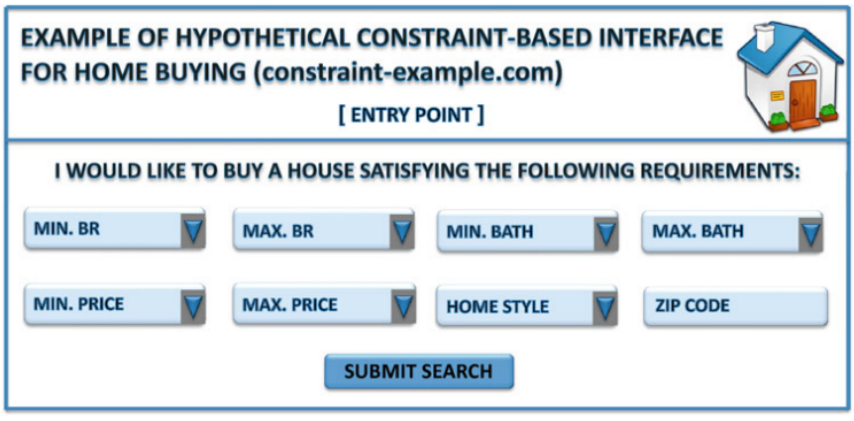


Figure 1.5: A hypothetical example of an initial user interface for a constraint-based recommender)

  2. *Case-based recommender systems* : user가 특정 case를 sepcified하고, 이를 이용하여 item attributes와 유사성을 계산한다. 여기에는 domain 지식을 사용하

여 domain-specific way로 정의된 similarity metrics가 사용된다. user는 반환 결과를 통해 새로운 target cases를 생성하며 상호작용의 과정을 거친다.

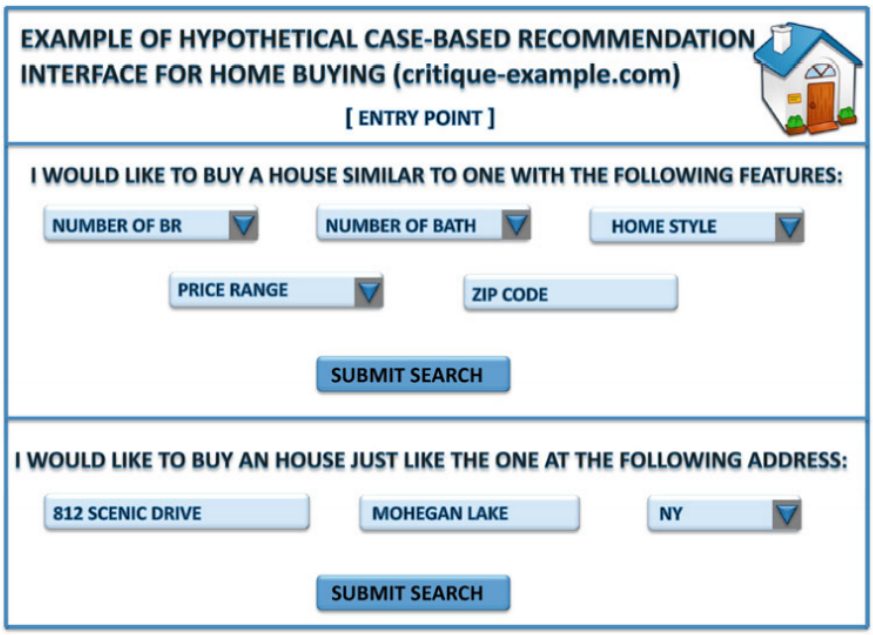


Figure 1.6: A hypothetical example of an initial user interface for a case-based recommender)

- knowledge-based recommender system이 반복적으로 추천을 하는 방법
  1. Conservation systems : user의 선호는 반복적인 feedback loop의 context에서 결정된다. item domain은 너무 복잡하고 user의 선호는 반복적 conservational system context에서 결정될 수 있다고 가정하기 때문이다.
  2. Search-based systems : 미리 정해진 질문의 순서를 사용하여 user의 선호가 결정된다.
     - ex) 시골/도시 어떤 곳에 있는 걸 선호하니?
  3. Navigation-based recommendation : user는 현재 추천된 item에서 변화시킬 요구사항을 기술한다.
     - ex)추천된 집과 비슷하지만 욕실이 하나 더 있으면 좋겠어.

- item의 content-attributes를 사용한다는 점에서 content-based systems과 비슷하다. 단지 cb는 past user behavior을, knowledge-based systems은 specification of their needs and interests를 이용한다는 점에서 다르다.
  - 따라서 cb와 마찬가지로 obvious한 예측을 한다는 단점이 있다.

1.3.3.1 Utility-Based Recommender Systems

- 상품의 특징에 기반하여 utility function을 정의하고 user가 item을 좋아할 가능성을 계산한다. utility value는 *priori(선험적)*으로 알려진 함수에 기반한다. 때문에 이러한 함수를 외부 지식이라고 바라보고, utility-based systems은 knowledge-based systems의 특수한 case라고도 할 수 있다.

1.3.5 Hybrid and Ensemble-Based Recommender Systems

- Content-based, Collaborative filtering, Knowledge-based, demographic systems을 모두 input sources로 활용하는 방법이다. 각 systems의 장점과 단점이 어우러져 좋은 system을 만들 수 있다. 이러한 ensemble을 통해 robust하고 성능이 좋은 model을 만들 수 있다. 또한 다른 systems의 ensemble 뿐만 아니라, 특정 systems(e.g., cf) 의 여러 models을 ensemble할 수도 있다.

1.3.6 Evaluation of Recommender Systems

- rating prediction의 경우 classification&regression의 evaluation을 사용하고, ranking prediction의 경우 search and information retrieval applications에서의 retrieval effectiveness의 evaluation을 사용한다.

1.6 Summary

- 추천시스템에는 다양한 알고리즘이 존재하고, 각 알고리즘 마다 장단점이 존재한다.
- 다양한 domain에서 추천 문제를 연구하고 있고, 다양한 유형의 input data와 knowledge bases가 존재한다.